include::attributes.txt[]

// Attributes

[.topic] [#ref-cli-cmd-init] = cdk init :keywords: {aws} CDK, {aws} CDK CLI, CDK Toolkit, IaC, Infrastructure as code, {aws} Cloud, {aws} CloudFormation, cdk init

== [abstract]

## Create a new {aws} CDK project from a template.

// Content start

Create a new {aws} CDK project from a template.

[#ref-cli-cmd-init-usage] == Usage

== [source,none,subs=“verbatim,attributes”]

$ cdk init ++++++++++++—-++++++++++++

[#ref-cli-cmd-init-args] == Arguments

[#ref-cli-cmd-init-args-template-type] *Template type*:: The CDK template type to initialize a new CDK project from. + –

* app – Template for a CDK application.
* lib – Template for an {aws} Construct Library.
* {blank}
* == sample-app – Example CDK application that includes some constructs.
* *Valid values*: app, lib, sample-app

[#ref-cli-cmd-init-options] == Options

For a list of global options that work with all CDK CLI commands, see xref:ref-cli-cmd-options[Global options].

[#ref-cli-cmd-init-options-generate-only] --generate-only <BOOLEAN>:: Specify this option to generate project files without initiating additional operations such as setting up a git repository, installing dependencies, or compiling the project. + *Default value*: false

[#ref-cli-cmd-init-options-help] --help, -h <BOOLEAN>:: Show command reference information for the cdk init command.

[#ref-cli-cmd-init-options-language] --language, -l <STRING>:: The language to be used for the new project. This option can be configured in the project’s cdk.json configuration file or at ~/.cdk.json on your local development machine. + *Valid values*: csharp, fsharp, go, java, javascript, python, typescript

[#ref-cli-cmd-init-options-list] --list <BOOLEAN>:: List the available template types and languages.

[#ref-cli-cmd-init-examples] == Examples

[#ref-cli-cmd-init-examples-1] === List the available template types and languages

== [source,none,subs=“verbatim,attributes”]

$ cdk init –list Available templates:

* app: Template for a CDK Application └─ cdk init app –language=[csharp|fsharp|go|java|javascript|python|typescript]
* lib: Template for a CDK Construct Library └─ cdk init lib –language=typescript
* sample-app: Example CDK Application with some constructs └─ cdk init sample-app –language=[csharp|fsharp|go|java|javascript|python|typescript] —

[#ref-cli-cmd-init-examples-2] === Create a new CDK app in TypeScript from the library template

== [source,none,subs=“verbatim,attributes”]

## $ cdk init lib –language=typescript